Shihong Huang*, Ivar Jacobson **, Panwei Ng *, Arne J. Berre****and Mira Kajko-Mattsson*******

*Associate Professor Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic Univeristy, USA
**Chairman, Ivar Jacobson International
***Asia Pacific CTO, Ivar Jacobson International
****Chief scientist, SINTEF and Associate Professor II, University of Oslo, Norway
*****Associate Professor, KTH Royal Institute of Technology

Essence – A Foundation for Software Development Games

Abstract— To be successful in software development, software teams must have the knowledge to systematically evaluate the progress and health of their projects, and detect and resolve risks early. How do teams acquire and apply such knowledge? How do teams adapt this knowledge to different development contexts? Essence is the result of the global SEMAT initiative that has taken place for a few years and has been adopted as a standard by the OMG in April 2014. Essence provides an innovative and novel user experience based on cards and game boards that are used to assess the progress and health of software development. Through the cards and the boards, developers can enact various development games, such as planning their sprints/iterations, agreeing on lifecycle models, and evaluating health and progress of their projects. Essence is an effective approach both in real software development and in software engineering education. Moreover, Essence provides the foundation for software engineering research and has been demonstrated as a framework for presenting case studies. This paper describes how Essence— the software engineering kernel and language—addresses these challenges and summarizes the tutorial given at ICSE 2014.

***Index Terms*—Essence, SEMAT, software engineering, kernel, practices, methods, agile, separate of concerns**

Introduction

Although software development endeavors vary with respect to their complexity, domain, novelty, and distribution, they all share a common ground of timeless principles and approaches underlying the ways of working. Such common ground is often hidden in the existing methods and it takes different forms that often practitioners, teams and academics are not aware of. Even large organizations that invest significant resources into defining their methods have difficulties to uncover the common ground[8].

A common ground could provide an excellent and stable foundation aiding developers in a couple of ways: (1) in understanding the core and timeless concepts and principles of software engineering, and (2) in reusing them across various software engineering endeavors. To uncover such common ground and its elements, the initiative SEMAT – Software Engineering Method and Theory – has been launched[4][5][6]. So far, it has succeeded to identify a common kernel of software engineering elements. Recently, this kernel has been endorsed to become an OMG standard [3].

The kernel[1][2] captures the essence of software engineering. Together with the language supporting the kernel, it constitutes what is called the Essence. The kernel includes a stripped-down, light-weight set of elements that are universal to all software engineering endeavors. Through the states defined for its elements, the Essence kernel constitutes an important instrument for reasoning about the progress and health of a software development endeavor. For a team, the kernel provides a conceptual basis and a roadmap for defining and evaluating the quality of its method. It also helps them understand where they are, point out what they should do next and, suggest what they should improve and where.

This tutorial introduces the Essence ideas and approach and demonstrates its practical applications. The goal is to help the audience to gain understanding of the basic concepts of the kernel and demonstrate how to use them in practice. The remainder of this paper is structured as follows. Section II briefly introduces the basics of Essence. Section III motivates why Essence is a paradigm shift. Section IV presents the contents of the tutorial. Finally, Section V summarizes the tutorial and points out the benefits of attending it.

Basics of Essence – Alpha and States

Essence provides a common ground for understanding and describing the commonalities and diversities of software engineering practices. Its kernel identifies a core set of elements called alphas, where each alpha represents a key dimension of the endeavor that needs to be progressed. As shown in Fig. 1 the alphas are *Opportunity*, *Stakeholders*, *Requirements*, *Software System*, *Team*, and *Work and Way of Working*. They are separated into three different areas of concerns, which are *Customer, Solution and Endeavor*.

Each alpha has a series of state progressions where each state is described with a detailed checklist specifying the criteria that need to be satisfied in order to progress to a particular state. In this way, Essence helps development teams understand the progress of each alpha and identify risks and challenges for that alpha. For example, as illustrated in Fig. 2, the set of states of requirements are *Conceived, Bounded, Coherent, Acceptable, Addressed,* and *Fulfilled*. While working on the requirements, the teams should progress these states in the above-listed order. If, for some reason they do not, they may encounter various risks. For example, if teams attempt to make requirements *Coherent* before getting them *Bounded* (i.e. understanding the boundaries), then they may risk wasting their effort on the unbounded requirements, and thereby, waste their time and resources on changing the boundaries during later phases of the development effort.

The kernel is actionable and extensible. The team can act on it and do something concrete with it. The team can also extend the kernel with further practices when needed. As illustrated in Fig. 3, both the

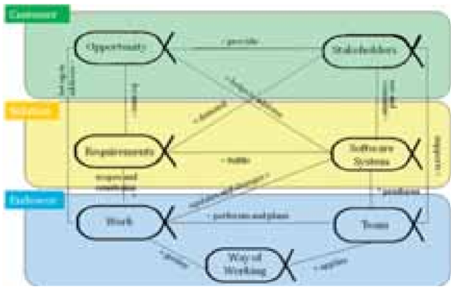

Fig. 1: Essence alphas and states

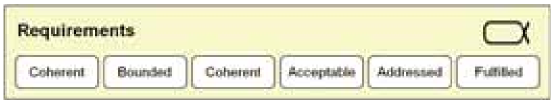

Fig. 2: Requirements alphas and states

kernel alphas and their states can be represented as a deck of cards. These cards may be used for monitoring the progress and health of software development endeavors. The monitoring may also be supported by spider graphs illustrating the common progress of all alphas.

Essence Represents a Paradigm Shift

This section explains why Essence is a paradigm shift, lists Essence's distinctive features, and finally, it discusses why it works hand-in-hand with the current agile trends.

Why a Paradigm Shift?

Essence is an innovative approach to managing software engineering endeavors. It is a paradigm shift implying a change in the approach of our ways of working. It shifts the focus from describing to doing. Traditionally, method frameworks such as RUP, SPEM, SME have focused on assisting method engineers in creating method descriptions. However, the adoption of these descriptions among software professionals has been very low. This is because these method descriptions are not effective enough in supporting development teams in their development efforts.

Being of a superlight narrative nature, Essence puts its focus on supporting the development teams in their ongoing work. It does not only assist them in mapping out what needs to be done, but it also helps them to achieve concrete and measurable results. It does this through a novel state-based, method-independent approach. Essence helps the teams to evaluate progress and health in a holistic manner in the most basic dimensions of a project such as, for instance, requirements, software system, team and other dimensions.

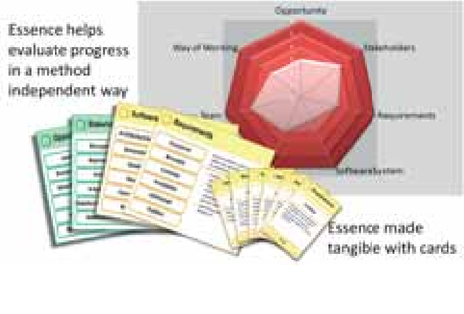
Essence helps evaluate progress in a method independent way
Opportunity
Way of Working
Stakeholders
Team
Requirements
Software System
Essence made tangible with cards

Fig. 3: Essence Pays Attention to Concrete Syntax

Distinctive features of Essence

Besides what have been discussed above, Essence also differs from traditional methods. It focuses on three distinctive features: (1) it relies on the separation of concerns, (2) it focuses on the agile principle, and (3) it considers the states of software engineering endeavors. Below, we briefly describe these three features.

Essence heavily relies on the "separation of concerns" principle. It separates the kernel from practice, governance and lifecycle specifics. It allows a method to be assembled from distinct and precise practices. This makes it easier for students and professionals to learn and grasp the fundamentals of software development and to grow in their understanding practice by practice. For teams and organizations, this permits safe and evolutionary approach to improve their ways of working.

Essence relies on the agile principle while working with methods. So far, the job of describing methods has been left to process engineers, resulting in a false dichotomy of describers and doers. With Essence, professionals are equipped and empowered to refine and improve their methods, that is, *describers* become *doers* and vice versa.

Special attention has been put to the design of Essence syntax. Essence focuses on the states of software engineering endeavors that are always relevant irrespective of the methods being used. All the Essence alphas include the states indicating their health and progress. In addition, the states are made concrete through a deck of state cards. The cards can be used in a variety of ways in different activities ranging from planning work, to assigning work, to defining work procedures, and to finalizing the work.

The above-described features cannot be achieved by simply extending previous and existing works. It takes a fundamental change in the way we work with methods and processes.

Essence works hand-in-hand with agile

No other process or method initiative has drawn such attention from the developer community as agile. The agile movement has provided a new way of looking at how we conduct the day-to-day activities of software development, how we build our teams and how we organize our work. Agile has led to the empowerment of the developers and the prominence of agile practices (such as Scrum and Test Driven Development) as the modern developer's practices of choice.

Essence and the SEMAT community provide a new way of approaching the domain of software engineering, a new way of understanding the progress and health of our development efforts, and a new way of combining practices into effective ways-of-working. This adds to the power of agility by adding a light however still a systematic and disciplined way of dealing with knowledge and putting this knowledge into practice[7].

The Tutorial Contents

This tutorial targets practitioners, as the SEMAT kernel is primarily focused on identifying the core and timeless concepts and principles underlying their ways of working. The tutorial also targets academics—both instructors and researchers who wish to systematically describe software engineering instead of presenting individual methods. This tutorial is structured in five parts:

1. *Introduction and Background of the Essence Kernel*: Discussing currently perceived problems and suggestions for remedying them by using Essence.
2. *Illustration of Using the Kernel*: Presenting the kernel, the alphas and alpha states and illustrating how they can be used in practice. This section includes hands on exercises practicing the use of the kernel and its elements. It provides hands-on experience on the kernel usage where the audience will get an opportunity to evaluate development endeavors in both agile and traditional contexts. Using tools, it will demonstrate how to do a project in iterations or sprints and how to continuously measure its progress. The details include: a) understanding the context; b) determining the current state; c) planning with Essence; and d) agreeing on the Lifecycle.
3. *The Essence Kernel and Language*: Describing the constitution of Essence Kernel, Alphas and Alpha states, and briefly introducing the language used to describe the Kernel.
4. *Case Studies*: Demonstrating some representative case studies of applying Essence in different areas, including:

a) Agile development b) Agile organizations and practices c) Essence Tooling.

5. *Ongoing and Future Work with Essence and SEMAT*: Presenting the ongoing work within the SEMAT community, including education[9], research[10] and certification process. Future directions of applying Essene are also presented.

Summary

Essence presents a paradigm shift in software engineering from describing to doing. The tutorial presents and explains Essence and its structure and demonstrates how Essence can be effectively used in practice. The kernel captures the common ground of software engineering, it is agile and lightweight easy to use. The tutorial's target groups are the practitioners in the industry and instructors and researchers in the academia. It focuses on supporting the usage of methods rather then defining them. The alpha and alpha states make it tangible to monitor the progress and health of software development. These are the critical features that separate Essence from previous and existing efforts in this space. Since the ideas are new and substantively different from what have been done in the past, the tutorial is very hands-on and interactive. Its participants will gaind good understanding of how to use Essence and, hopefully, they will getmotivated for adopting it in their daily work.

References

[1] Jacobson, I., Ng, P-W., McMahon, P., Spence, I., Lidman, S., *The Essence of Software Engineering—Applying the SEMAT Kernel*. Addison-Wesley. Jan. 2013

[2] Jacobson, I., Ng, P-W., McMahon, P., Spence, I., Lidman, S., The Essence of Software Engineering: The SEMAT Kernel, *Communications of the ACM*, 55(12), 2012

[3] Essence–Kernel and Language for Software Engineering Methods, http://www.omg.org/spec/Essence/1.0/Beta1/ July 2013

[4] Jacobson, I., Huang, S., Kajko-Mattsson, M., McMahon, P., Seymour, E., "SEMAT: Three Year Vision." *Programming and Computer Software* 38(1):1-12, Springer 2012

[5] Jacobson, I., Meyer, B., Soley, R. The SEMAT Vision Statement http://blog.paluno.uni-due.de/semat.org/wp-content/uploads/2012/03/SEMAT-vision.pdf

[6] SEMAT Website: www.semat.org

[7] Jacobson, I., Spence, I., Ng, P-W., Agile and SEMAT— Perfect Partners. *ACM Queue*. November 5, 2013

[8] Jacobson, I., Ng, P-W., Spence, I., McMahon, P. Major-league SEMAT: Why Should an Executive Care? *ACM Queue*, February 27, 2014

[9] Ng, P-W., Huang, S., Essence: A Framework to Help Bridge the Gap between Software Engineering Education and Industry Needs. *Proceedings of the IEEE 26th Conference on Software Engineering Education and Training (CSEE&T)*, Co-located with ICSE 2013. May 19-21, 2013. San Francisco, CA, USA. pp. 304 – 308.

[10] Ng, P-W., Huang, S., Wu., Y On the Value of Essence to Software Engineering Research: A Preliminary Study. *Proceedings of 2nd SEMAT Workshop on a General Theory of Software Engineering (GTSE 2013)*, co-located with ICSE 2013. May 26, 2013, San Francisco, CA, USA. pp. 51 – 58.

About the Authors

Shihong Huang is an Associate Professor in the Department of Computer & Electrical Engineering and Computer Science at Florida Atlantic Univeristy, USA. Her current research interests include requirements engineering, software systems modeling and human computer interaction. Her earlier work dealt with reverse engineering, program comprehension, and software system redocumentation. She has co-organized many workshops and panel discussions in international conferences. She was the General Chair of the 24th ACM International Conference on Design of Communication, and was Program Co-Chair of the 9th IEEE International Symposium on Web Site Evolution. She is one of the main contributors of the OMG Essence standardization.

DR. Ivar Jacobson, the chairman of Ivar Jacobson International, is a father of components and component architecture, use cases, the Unified Modeling Language, and the Rational Unified Process. He has contributed to modern business modeling and aspect-oriented software development. He is an international honorary advisor at Peking University, Beijing, and holds an honorary doctorate from San Martin de Porres University, Peru.

DR. Pan-Wei Ng, the Asia Pacific CTO at Ivar Jacobson International, coaches large-scale systems development involving many millions of lines of code and hundreds of people per release, helping them transition to a lean and agile way of working, not forgetting to improve their code and architecture and to test through use cases and aspects. He is the coauthor, with Ivar Jacobson, of Aspect-oriented Software Development with Use Cases. He believes in making things tangible and practical and has been an active contributor to ideas behind the kernel, including the use of state cards.

Arne J. Berre is a chief scientist at SINTEF and associate professor II at the University of Oslo. He is engaged in European research projects on methods and architectures for interoperable software and services, and in standardization activities at OMG, including the lead of the Essence, SoaML and VDML standardization activities. Dr. Berre has organized tutorials, workshops, and sessions at software technology venues such as ECOOP, OOPSLA and WICSA.

Mira Kajko-Mattsson is Associate Professor in software engineering at *School of Information and Communication Technology* at *KTH Royal Institute of Technology.* Her expertise lies in industrial lifecycle software processes. The primary objective of her research is to educe theory about various lifecycle processes, put this theory into process models, evaluate the effects of changing technologies and working patterns on the lifecycle processes and make suggestions for their improvement.

Kajko-Mattsson is the authour of 170 international peer-reviewed publications in highly respected conferences and journals. She has given four keynote speeches and received 13 international awards. Presently, she is an active member of SEMAT as a co-creator and co-founder of software engineering theory.

Ian Spence is CTO at Ivar Jacobson International and the team leader for the development of the SEMAT kernel. An experienced coach, he has introduced hundreds of projects to iterative and agile practices. He has also led numerous successful large-scale transformation projects working with development organizations of up to 5,000 people. His current interests are agile for large projects, agile outsourcing, and driving sustainable change with agile measurements.

About the Authors

Paul E Mcmahon (pemcmahon@acm.org) is an independent consultant focusing on coaching project managers, team leaders, and software professionals in the practical use of lean and agile techniques in constrained environments. He is a Certified ScrumMaster and a Certified Lean Six Sigma Black Belt. He has been a leader in the SEMAT initiative since its inception.

Announcement of Tutorial on Essence – A Foundation for Software Development Games

Collocated with the 36th International Conference on Software Engineering (Hyderabad International Convention Centre, Hyderabad, India)

SEMAT

This tutorial is on the new OMG standard "Essence — the software engineering kernel and language

Target Audience

Managers, leaders, team members, and agile change agents who wants a responsible for empowering software teams of different kinds to become better, faster, cheaper and happier

Essence of Software development in a deck of cards

Tutorial Time

Tuesday June 3, 2014
9:00am – 12:30pm

Tutorial Venue

Hyderabad International Convention Centre, Hyderabad, India (Collocated with the 36th International Conference on Software Engineering)

Presenters

Shihong Huang Associate Professor, Florida Atlantic University, USA
Ivar Jacobson Chairman, Ivar Jacobson International, Switzerland
Pan-Wei Ng Asia Pacific CTO, Ivar Jacobson International, China
Mira Kajko-Mattsson Associate Professor, Royal Institute of Technology, Sweden
Arne J. Berre Chief scientist, SINTEF Oslo, Norway

Tutorial Description

To become successful in software development, software teams must have the knowledge to systematically evaluate the progress and health of their projects, and detect and resolve risks early. How do teams acquire and apply such knowledge? How do teams adapt this knowledge to different development contexts? This tutorial demonstrates how Essence, the software engineering kernel and language, addresses these challenges.

Essence is the result of the global SEMAT initiative, (see www.semat.org), that has taken place for a few years and now recently been adopted as a standard by the OMG (http://www.omg.org/spec/Essence/). Essence provides an innovative and novel user experience based on cards and game boards that are used to assess the progress and health of software development. Through gamification with cards and boards developers can enact various development games, such as planning sprints/iterations, agreeing on lifecycle models, evaluating health and progress of a project. Participants will gain hands-on experience with the cards and games in this highly interactive and engaging tutorial.

This tutorial also introduces some real world case studies in which Essence cards and games are applied. We also demonstrate how to use Essence as a foundation for reporting and evaluating software engineering research.

Tutorial Outline

1. Introduction and Background of the Essence Kernel
2. Illustration of Using the Kernel
   a. Understanding the Context
   b. Determine the Current State
   c. Planning with Essence
   d. Agreeing the lifecycle
3. The Essence Kernel and Language
4. Case Studies
   a. Agile development
   b. Agile organizations and practices
   c. Essence Tooling
5. Ongoing and Future Work with Essence and SEMAT

Registration

Indian Delegates Registration Link (Special offer for Indian/local participants) http://events.kwconferences.com/ei/ICSE2014/registrations.htm

International Delegation Registration:
https://regmaster4.com/2014conf/ICSE14/register.php